

I. REAL PARTY IN INTEREST	1
II. RELATED APPEALS AND INTERFERENCES	1
III. STATUS OF CLAIMS.....	2
IV. STATUS OF AMENDMENTS	2
V. SUMMARY OF CLAIMED SUBJECT MATTER.....	2
VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL.....	5
VII. THE ARGUMENT	5
VIII. CLAIMS APPENDIX	14
IX. EVIDENCE APPENDIX	20
X. RELATED PROCEEDINGS APPENDIX	21

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

Application Number: 10/650,894
Filing Date: 8/28/2003
Applicant(s): Jason Robert McGee
Entitled: SYSTEM AND METHOD FOR PROVIDING
SHARED WEB MODULES
Examiner: Kevin T. Bates
Group Art Unit: 2456
Attorney Docket No.: RSW920030102US1 (7161-508U)

TRANSMITTAL OF APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Submitted herewith is Appellants' Appeal Brief in support of the Notice of Appeal filed April 23, 2009. Please charge any shortage in fees due under 37 C.F.R. §§ 1.17, 41.20, and in connection with the filing of this paper, including extension of time fees, to Deposit Account 09-0461, and please credit any excess fees to such deposit account.

Date: June 23, 2009

Respectfully submitted,

/Steven M. Greenberg/
Steven M. Greenberg, Registration No. 44,725
Customer Number 46320

PATENT

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

Application Number: 10/650,894
Filing Date: 8/28/2003
Applicant(s): Jason Robert McGee
Entitled: SYSTEM AND METHOD FOR PROVIDING
SHARED WEB MODULES
Examiner: Kevin T. Bates
Group Art Unit: 2456
Attorney Docket No.: RSW920030102US1 (7161-508U)

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal filed April 23, 2009, wherein Appellants appeal from the Examiner's rejection of claims 1, 2, 4 through 12 and 14 through 20.

I. REAL PARTY IN INTEREST

This application is assigned to International Business Machines Corporation by assignment recorded on August 28, 2003 at Reel 014464, Frame 0318.

II. RELATED APPEALS AND INTERFERENCES

Appellant is unaware of any related appeals and interferences.

III. STATUS OF CLAIMS

Claims 1, 2, 4 through 12 and 14 through 20 are pending in this Application and have been four times rejected. Claims 3 and 13 were canceled in the Amendment dated November 19, 2008 (the "Amendment"). It is from the multiple rejections of claims 1, 2, 4 through 12 and 14 through 20 that this Appeal is taken.

IV. STATUS OF AMENDMENTS

Claims 1, 7, 11, 12 and 14 through 20 were amended in the Amendment.

V. SUMMARY OF CLAIMED SUBJECT MATTER

By reference to page 4, lines 3 through 24 Appellants' originally filed specification, Appellants have invented a system and method for sharing web modules amongst different web applications. With the system and method of Appellants' invention, shared web modules are stored in an archive data structure and are accessible for use with web applications. The web applications, and/or web modules of the web applications, may include a shared web module designation file that identifies the shared web modules that are to be incorporated into the web modules for the web application. Notably, a runtime component is provided that determines, whether the web application and/or web modules have a shared web module designation file that identifies shared web modules that are to be incorporated into the web modules of the web application. If so, the runtime component locates these shared web modules and logically

merges them with the web module of the web application taking into account precedence or priority of the shared web modules as designated by the shared web module designation file. The result is a logically merged web module that contains the shared web modules referenced in the shared web module

With specific reference to claim 1, a method of generating a logically merged web module for a web application is provided. (Page 25, Lines 1-2) The method includes identifying, in response to a determination that a shared module designation file exists, at least one shared web module from the shared module designation file to be incorporated into a web application, in order to form at least one identified shared web module. (Page 26, Lines 3-17) Of note, the shared web module designation file includes all descriptors that reference the shared web module. (Page 20, Lines 1-7) The method also includes locating the identified shared web module using path information. (Page 26, Lines 17-20) Finally, the method includes logically merging the shared web module with web modules of the web application, in accordance with the shared web module designation file to generate a logically merged web application. (Page 26, Lines 21-28) Consequently, a reference to the shared web module is used in the logically merged web application rather than a copy of the at least one shared web module. (Page 21, Lines 21-24)

With specific reference to claim 11, a computer program product is provided in a computer readable medium for generating a logically merged web module for a web application. (Page 28, Lines 6-9) The computer readable medium includes computer executable instructions that, in response to a determination that a shared module designation file exists, identifies at least

one shared web module from the shared module designation file to be incorporated into a web application, to form at least one identified shared web module, (Page 26, Lines 3-17) such that the shared web module designation file includes all descriptors that reference the at least one shared web module. (Page 20, Lines 1-7) The computer readable medium further includes computer executable instructions for locating the shared web module locating the identified shared web module using path information. (Page 26, Lines 17-20) Finally, computer readable medium finally includes computer executable instructions for logically merging the shared web module with web modules of the web application, in accordance with the shared web module designation file, to generate a logically merged web application. (Page 26, Lines 21-28) Consequently, a reference to the shared web module is used rather than a copy of the shared web module in the logically merged web application. (Page 21, Lines 21-24)

With specific reference to claim 19, an apparatus for generating a logically merged web module for a web application is provided to include a system bus, a local memory connected to the system bus, wherein the memory contains computer executable instructions and a processor connected to the system bus. (Page 9, Line 25 through Page 10, Lines 1-4) The process can executes the computer executable instructions to direct the apparatus, in response to a determination that a shared module designation file exists, to identify at least one shared web module from the shared module designation file to be incorporated into a web application. (Page 26, Lines 3-17) The process also can execute the computer executable instructions to direct the apparatus to form at least one identified shared web module, (Page 26, Lines 3-17) wherein the shared web module designation file includes all descriptors that reference the at least one shared web module. (Page 20, Lines 1-7) The process further can execute the computer executable

instructions to direct the apparatus to locate the identified shared web module using path information. (Page 26, Lines 17-20) Finally, the process can execute the computer executable instructions to direct the apparatus to logically merge the shared web module with web modules of the web application, in accordance with the shared web module designation file to generate a logically merged web application, (Page 26, Lines 21-28) wherein a reference to the at least one shared web module is used rather than a copy of the at least one shared web module in the logically merged web application. (Page 21, Lines 21-24)

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1, 2, 4, 6 through 12 and 14 through 20 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,519,594 to Li in view of U.S. Patent Application Publication No. 2004/0255293 by Spotswood.

Claim 5 has been rejected under 35 U.S.C. § 103(a) as being unpatentable over Li in view of Spotswood and further in view of U.S. Patent No. 6,721,777 to Sharma.

VII. THE ARGUMENT

THE REJECTION OF CLAIMS 1, 2, 4, 6 THROUGH 12 AND 14 THROUGH 20 UNDER 35 U.S.C. § 103

For convenience of the Honorable Board in addressing the rejections, claims 2, 4 and 6 through 10 stand or fall together with claim 1, and claims 12 and 14 through 20 stand or fall together with claim 11.

“[T]he examiner bears the initial burden, on review of the prior art or on any other ground, of presenting a prima facie case of unpatentability.”¹ A rejection under 35 U.S.C. § 103(a) must be based on the following factual determinations: (1) the scope and content of the prior art; (2) the level of ordinary skill in the art; (3) the differences between the claimed invention and the prior art; and (4) any objective indicia of non-obviousness.² Appellants’ position is that the Examiner has not properly established the underlying facts regarding (1) the scope and content of the prior art and (3) the differences between the claimed invention and the prior art.

Specifically, on October 10, 2007, the Patent Office issued the "Examination Guidelines for Determining Obviousness Under 35 U.S.C. 103 in View of the Supreme Court Decision in KSR International Co. v. Teleflex Inc.," 73 Fed. Reg. 57,526 (2007) (hereinafter the Examination Guidelines). Section III is entitled "Rationales To Support Rejections Under 35 U.S.C. 103." Within this section is the following quote from the Supreme Court: "rejections on obviousness grounds cannot be sustained by merely conclusory statements; instead there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." KSR Int'l Co., 127 S. Ct. 1727, 1741 (2007) (quoting In re Kahn, 441 F.3d 977, 988 (Fed. Cir. 2006)).

¹ In re Oetiker, 977 F.2d 1443, 1445 (Fed. Cir. 1992).

² See KSR Int'l Co. v. Teleflex Inc., 127 S.Ct. 1727, 1734, 82 USPQ2d 1385, 1391 (2007); DyStar Textilfarben GmbH & Co. Deutschland KG v. C.H. Patrick Co., 464 F.3d 1356, 1360 (Fed. Cir. 2006) (citing Graham v. John Deere Co., 383 U.S. 1, 17 (1966)).

Referring to the first column on page 57,529 of the Examination Guidelines, the following is a list of rationales that may be used to support a finding of obviousness under 35 U.S.C. § 103:

- (A) Combining prior art elements according to known methods to yield predictable results;
- (B) Simple substitution of one known element for another to obtain predictable results;
- (C) Use of known technique to improve similar devices (methods, or products) in the same way;
- (D) Applying a known technique to a known device (method, or product) ready for improvement to yield predictable results;
- (E) "Obvious to try" - choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success;
- (F) Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations would have been predictable to one of ordinary skill in the art;
- (G) Some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

Upon reviewing the Examiner's analysis in the paragraph spanning pages 2 and 3 of the Final Office Action, the Examiner appears to be employing rationale (A). If the Examiner is not relying upon rationale (A), Appellants request that the Examiner clearly identify the rationale, as described in the Examination Guidelines, being employed by the Examiner in rejecting the claims under 35 U.S.C. § 103.

The Examination Guidelines for rationale (A) set forth a precise process for which the Examiner must follow in order to establish a prima facie case of obviousness under 35 U.S.C. § 103(a). Specifically, to reject a claim based on this rationale, Office personnel must resolve the Graham factual inquiries. Thereafter, Office personnel must then articulate the following:

- (1) a finding that the prior art included each element claimed, although not necessarily in a single prior art reference, with the only difference between the claimed invention and the prior art being the lack of actual combination of the elements in a single prior art reference;
- (2) a finding that one of ordinary skill in the art could have combined the elements as claimed by known methods, and that in combination, each element merely would have performed the same function as it did separately;
- (3) a finding that one of ordinary skill in the art would have recognized that the results of the combination were predictable; and

(4) whatever additional findings based on the Graham factual inquiries may be necessary, in view of the facts of the case under consideration, to explain a conclusion of obviousness.

With respect, Examiner has not adequately articulated a finding that the prior art included each element claimed with the only difference between the claimed invention and the prior art being the lack of actual combination of the elements in a single prior art reference.

Further, as part of the analysis, Examiner must properly construe each critical claim term present in the claims before performing a comparison of the cited art to the claims in establishing a finding that the cited art included each element claimed. More specifically, obviousness under § 102 is a two-step inquiry. The first step is a **proper construction of the claims**. ... The second step requires a comparison of the **properly construed claim** to the prior art.”³ During patent examination, the pending claims must be “given their broadest reasonable interpretation consistent with the specification,”⁴ and the broadest reasonable interpretation of the claims must also be consistent with the interpretation that those skilled in the art would reach.⁵ Therefore, the Examiner must (i) identify the individual elements of the claims and properly construe these individual elements,⁶ and (ii) identify corresponding elements disclosed in the allegedly anticipating reference and compare these allegedly corresponding elements to the individual elements of the claims.⁷ This burden has not been met as well.

³ Medichem, S.A. v. Rolabo, S.L., 353 F.3d 928, 933 (Fed. Cir. 2003) (internal citations omitted).

⁴ In re ICON Health and Fitness, Inc., 496 F.3d 1374, 1379 (Fed. Cir. 2007) (“[T]he PTO must give claims their broadest reasonable construction consistent with the specification. Therefore, we look to the specification to see if it provides a definition for claim terms, but otherwise apply a broad interpretation.”); In re Hyatt, 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed. Cir. 2000).

⁵ In re Cortright, 165 F.3d 1353, 1359, 49 USPQ2d 1464, 1468 (Fed. Cir. 1999)

⁶ See also, Panduit Corp. v. Dennison Mfg. Co., 810 F.2d 1561, 1567-68 (Fed. Cir. 1987) (In making a patentability determination, analysis must begin with the question, “what is the invention claimed?” since “[c]laim interpretation... will normally control the remainder of the decisional process”); see Gechter v. Davidson, 116 F.3d 1454, 1460 (Fed. Cir. 1997) (requiring explicit claim construction as to any terms in dispute).

⁷ Lindermann Maschinenfabrik GMBH v. American Hoist & Derrick Co., 730 F.2d 1452, 221 USPQ 481 (Fed. Cir. 1984).

By way of example, claim 1 recites a method of generating a logically merged web module for a web application. For the convenience of the Honorable Board, claim 1 has been reproduced as follows:

1. A method of generating a logically merged web module for a web application, comprising:
 - responsive to a determination that a shared module designation file exists, identifying at least one shared web module from the shared module designation file to be incorporated into a web application, to form at least one identified shared web module, wherein the shared web module designation file includes all descriptors that reference the at least one shared web module;
 - locating the at least one identified shared web module using path information;
 - logically merging the at least one shared web module with web modules of the web application, in accordance with the shared web module designation file to generate a logically merged web application, wherein a reference to the at least one shared web module is used in the logically merged web application rather than a copy of the at least one shared web module.

Integral to claim 1 (and also claims 11 and 19) is the inclusion of all descriptors that reference the shared web module in the shared module designation file, and the logical merging of the shared web module with the web modules of the web application. On page 2 of the Final Office Action, Examiner performs an implicit claim construction of "descriptors" by comparing the limitation of "logically merging the at least one shared web module using path information" with column 9, lines 16 through 32 of Li.

For the convenience of the Honorable Board, the entirety of column 9, lines 16 through 32 of Li are reproduced herein as follows:

FIG. 8 illustrates physical and logical memory organization of the classes within the memory pool 280. To avoid the physical copying and moving of classes when operations are performed on the classes by multiple JVMs 251-253, these classes are stored in memory 310 in the manner as shown in FIG. 8. The class is stored in variable sized class cells or "blocks" 290-294. The actual class is constructed by linking cells together at certain offsets in the class cells as shown by 324a-324c. The size of any block can be dynamically adjusted based on the size of the stored Java class. The link contains the name of the class, its size and the offset from the start. The entries in the name table are shown as 322a-322c. A hash table 320 is used to reference the position within the name table for an enumerated Java class. The name is then used to reference an offset which points to the memory position of the class within the shared memory pool 280.

Thus, Examiner has implicitly construed descriptor as "link". The term descriptor is well-known amongst those skilled in the computing arts as a file--oftentimes a markup language specified file--that describes the contents or functionality of another object. Appellants' usage of the term "descriptor" is consistent with its ordinary meaning as evidenced by page 20, lines 1 through 7 reproduced herein as follows:

As touched on above, the shared web module designation **416** stores the descriptors of the shared web modules **422-426** that are to be logically merged with the web modules **412-414** of the web application **410**. These descriptors may include, for example, filename (and possibly the path) for the shared web module and a merge priority for the shared web module.

Examiner, however, in construing "descriptor" as a mere link in a name table referencing memory attributes for a Java class exceeds the ordinary boundaries set forth for claim construction in M.P.E.P. 2111.

More specifically, as set forth in M.P.E.P. 2111, "During patent examination, the pending claims must be given their broadest reasonable interpretation consistent with the specification." Specifically, the Federal Circuit's en banc decision in Phillips v. AWH Corp., 415 F.3d 1303, 75 USPQ2d 1321 (Fed. Cir. 2005) expressly recognized that the USPTO employs the "broadest reasonable interpretation" standard.⁸ Examiner's improper claim construction of "descriptor" as merely a link in a table exceeds the legal standard for claim construction during examination and inhibits Examiner's ability to properly compare the cited art to Appellant's claims. Examiner's improper claim construction of "descriptor" results in a failed comparison of the combination of Li and Spotswood under rationale (A) and represents clear reversible error.

⁸ The Patent and Trademark Office ("PTO") determines the scope of claims in patent applications not solely on the basis of the claim language, but upon giving claims their broadest reasonable construction "in light of the specification as it would be interpreted by one of ordinary skill in the art." In re Am. Acad. of Sci. Tech. Ctr., 367 F.3d 1359, 1364, 70 USPQ2d 1827 (Fed. Cir. 2004). Indeed, the rules of the PTO require that application claims must "conform to the invention as set forth in the remainder of the specification and the terms and phrases used in the claims must find clear support or antecedent basis in the description so that the meaning of the terms in the claims may be ascertainable by reference to the description." 37 CFR 1.75(d)(1).

Examiner has committed similar clear reversible error in connection with the critical claim term "logically merging". The term "logically merging" appears in each of claims 1, 11 and 19 and Examiner on page 3 of the Final Office Action provides an implicit claim construction for "logically merging" in comparing the limitations incorporating "logically merging" to column 10, lines 25 through 49 of Li. Specifically, Examiner stated,

logically merging the at least one shared module with modules of the web application, in accordance with the shared module designation file to generate a logically merged web application (Col. 10, lines 26 - 49), wherein a reference to the at least one shared web module is used in the logically merged application rather than a copy of the at least one shared module (Col. 8, lines 22 26).

Column 10, lines 26 through 49 of Li also are reproduced herein for the convenience of the Honorable Board:

The remainder of the steps of FIG. 10 illustrate the procedure used for accessing the stored class(N) that resides in the shared memory pool. At step 418, a JVM, e.g., JVM(J), checks with the JCM to determine whether or not a class(N) is stored in the shared memory pool. It is appreciated that JVM(I) and JVM(J) are both running simultaneously on system 112 (or could be the same JVM). The above determination is typically done by an instruction whose function is to establish class(N) for the JVM. During the establishment function, before the class is written into memory, a check is first done to determine if the class has already been established in the shared memory pool by another JVM. At step 420, if class(N) exists in the shared memory pool, then JVM(J) requests and obtains the read key for class(N) from the JCM. At step 422, JVM(J) then accesses class(N), as needed, from the shared memory pool instead of copying its own instantiation of class(N) in memory. At step 424, JVM(J) then releases the read key for class(N) when done with this class. It is appreciated that steps 418-424 can be repeated for other JVMs that are simultaneously running on computer system 112. Memory is thereby saved because each JVM that needs class(N) can share the instantiation of this class as established by JVM(I) without copying its own version.

Based upon Examiner's comparison, it is fair to presume that Examiner has construed "logically merging" to mean "sharing an instantiation". In the field of computer science the notion of merging means the combination of two objects, while the notion of a logical merge is the presentation of an object reflective of the merger of multiple other objects by logic only without creating a physically merged object. By comparison, the notion of sharing an instantiation literally means multiple different objects can utilize an instance of a class. Accordingly, Examiner's claim construction

makes no sense and again, results in a failed comparison of the combination of Li and Spotswood under rationale (A) and represents clear reversible error.

THE REJECTION OF CLAIM 5 UNDER 35 U.S.C. § 103

In as much as Examiner has failed to prove a prima facie case of obviousness with respect to independent claim 1, it naturally follows that Examiner has failed to prove a prima facie case of obviousness with respect to dependent claim 5 at least for the reasons set forth above.

Based upon the foregoing, Appellants respectfully believe that the Examiner's rejections under 35 U.S.C. §§ 102(b) and 103(a) are not viable and Appellants, therefore, respectfully solicit the Honorable Board to reverse the Examiner's rejections under 35 U.S.C. §§ 102(b) and 103(a).

To the extent the Examiner, having considered the foregoing arguments, persists and prepares an Examiner's Answer, Examiner is reminded of Examiner's responsibility under M.P.E.P. 1207.02(A)(1)(9)(e) to map every claim term in any of claims 1, 11 or 19 to the combination of the Li and Spotswood reference. In this regard, for the convenience of the Examiner the entirety of is provided herein:

For each rejection under 35 U.S.C. 102 or 103 where there are questions as to how limitations in the claims correspond to features in the prior art even after the examiner complies with the requirements of paragraphs (c) and (d) of this section, the examiner must compare at least one of the rejected claims **feature by feature** with the prior art relied on in the rejection. **The comparison must align the language of the claim side-by-side with a reference to the specific page, line number, drawing reference number, and quotation from the prior art, as appropriate.**

Specifically, Examiner must point out with particularity the precise teaching in Li that maps to the claimed elements (1) "descriptor"; and, (2) "logically merging", while applying the ordinary

meaning of "descriptor" and "logically merging " as a proper claim construction under M.P.E.P. 2111.01(I).⁹

Date: June 23, 2009

Respectfully submitted,

/Steven M. Greenberg/

Steven M. Greenberg

Registration No. 44,725

Customer Number 46320

⁹ Although claims of issued patents are interpreted in light of the specification, prosecution history, prior art and other claims, this is not the mode of claim interpretation to be applied during examination. During examination, the claims must be interpreted as broadly as their terms reasonably allow. In re American Academy of Science Tech Center, 367 F.3d 1359, 1369, 70 USPQ2d 1827, 1834 (Fed. Cir. 2004)

VIII. CLAIMS APPENDIX

1. (Previously Amended) A method of generating a logically merged web module for a web application, comprising:

responsive to a determination that a shared module designation file exists, identifying at least one shared web module from the shared module designation file to be incorporated into a web application, to form at least one identified shared web module, wherein the shared web module designation file includes all descriptors that reference the at least one shared web module;

locating the at least one identified shared web module using path information;

logically merging the at least one shared web module with web modules of the web application, in accordance with the shared web module designation file to generate a logically merged web application, wherein a reference to the at least one shared web module is used in the logically merged web application rather than a copy of the at least one shared web module.

2. (Original) The method of claim 1, further comprising:

loading the logically merged web application into a web container.

3. (Canceled)

4. (Original) The method of claim 1, wherein the web application is an enterprise archive (EAR) and wherein the logically merged web application is a logically merged EAR.

5. (Original) The method of claim 1, wherein the at least one shared web module includes at least one of a web archive (WAR) file, an enterprise java bean (EJB) archive file, and a resource archive (RAR) file.
6. (Original) The method of claim 1, wherein logically merging the at least one shared web module with web modules of the web application includes:
- determining a priority associated with the at least one shared web module; and
 - resolving any conflicts between shared web modules in the at least one shared web module and conflicts between the at least one shared web module and web modules of the web application, if any.
7. (Previously Amended) The method of claim 1, wherein the steps of identifying, locating and logically merging are performed during an initialization process of a runtime environment for initializing the web application to be run on a server.
8. (Original) The method of claim 1, wherein logically merging the at least one shared web module with the web modules of the web application includes using a service provider interface (SPI) that provides merge logic for merging different module types.
9. (Previously Amended) The method of claim 2, wherein the container uses one or more application program interfaces (APIs) to identify a path to the at least one shared web module and loads the at least one shared web module when loading the logically merged web application.

10. (Original) The method of claim 1, wherein logically merging the at least one shared web module with web modules of the web application includes at least one of relinking references to the at least one shared web module in the web modules of the web application, extrapolating policy information for the at least one shared web module from a policy file associated with the web application, and modifying a class path for the web application to include paths to each of the at least one shared web modules.

11. (Previously Amended) A computer program product in a computer readable medium for generating a logically merged web module for a web application, comprising:

computer executable instructions responsive to a determination that a shared module designation file exists, for identifying at least one shared web module from the shared module designation file to be incorporated into a web application, to form at least one identified shared web module, wherein the shared web module designation file includes all descriptors that reference the at least one shared web module;

computer executable instructions for locating the at least one shared web module locating the at least one identified shared web module using path information; and

computer executable instructions for logically merging the at least one shared web module with web modules of the web application, in accordance with the shared web module designation file, to generate a logically merged web application, wherein a reference to the at least one shared web module is used rather than a copy of the at least one shared web module in the logically merged web application.

12. (Previously Amended) The computer program product of claim 11, further comprising:

computer executable instructions for loading the logically merged web application into a web container.

13. (Canceled)

14. (Previously Amended) The computer program product of claim 11, wherein the computer executable instructions for logically merging the at least one shared web module with web modules of the web application include:

computer executable instructions for determining a priority associated with the at least one shared web module; and

computer executable instructions for resolving any conflicts between shared web modules in the at least one shared web module and conflicts between the at least one shared web module and web modules of the web application.

15. (Previously Amended) The computer program product of claim 11, wherein the computer executable instructions are executed during an initialization process of a runtime environment for initializing the web application to be run on a server.

16. (Previously Amended) The computer program product of claim 11, wherein the computer executable instructions for logically merging the at least one shared web module with

the web modules of the web application include computer executable instructions for using a service provider interface (SPI) that provides merge logic for merging different module resources.

17. (Previously Amended) The computer program product of claim 12, wherein the container uses one or more application program interfaces (APIs) to identify a path to the at least one shared web module.

18. (Previously Amended) The computer program product of claim 11, wherein the third instructions for logically merging the at least one shared web module with web modules of the web application include computer executable instructions for relinking references to the at least one shared web module in the web modules of the web application, computer executable instructions for extrapolating policy information for the at least one shared web module from a policy file associated with the web application, and computer executable instructions for modifying a class path for the web application to include paths to each of the at least one shared web modules.

19. (Previously Amended) An apparatus for generating a logically merged web module for a web application, comprising:

- a system bus;

- a local memory connected to the system bus, wherein the memory contains computer executable instructions;

a processor connected to the system bus, wherein the processor executes the computer executable instructions to direct the apparatus to:

responsive to a determination that a shared module designation file exists, identify at least one shared web module from the shared module designation file to be incorporated into a web application, to form at least one identified shared web module, wherein the shared web module designation file includes all descriptors that reference the at least one shared web module;

locate the at least one identified shared web module using path information; and

logically merge the at least one shared web module with web modules of the web application, in accordance with the shared web module designation file to generate a logically merged web application, wherein a reference to the at least one shared web module is used rather than a copy of the at least one shared web module in the logically merged web application.

20. (Previously Amended) The computer program product of claim 19, wherein the processor further executes the computer executable instructions to direct the apparatus to load the logically merged web application into a web container.

IX. EVIDENCE APPENDIX

No evidence submitted pursuant to 37 C.F.R. §§ 1.130, 1.131, or 1.132 of this title or of any other evidence entered by the Examiner has been relied upon by Appellant in this Appeal, and thus no evidence is attached hereto.

X. RELATED PROCEEDINGS APPENDIX

Since Appellant is unaware of any related appeals and interferences, no decision rendered by a court or the Board is attached hereto.